



TITLE:

On the Size of Ordered Binary Decision Diagrams Representing Threshold Functions

AUTHOR(S):

HOSAKA, Kazuhisa; TAKENAGA, Yasuhiko; YAJIMA, Shuzo

CITATION:

HOSAKA, Kazuhisa ...[et al]. On the Size of Ordered Binary Decision Diagrams Representing Threshold Functions. 数理解析研究所講究録 1994, 871: 87-93

ISSUE DATE:

1994-05

URL:

<http://hdl.handle.net/2433/84052>

RIGHT:

On the Size of Ordered Binary Decision Diagrams Representing Threshold Functions

Kazuhisa HOSAKA Yasuhiko TAKENAGA Shuzo YAJIMA

保坂和寿

武永康彦

矢島脩三

Department of Information Science, Faculty of Engineering, Kyoto University

Abstract

An ordered binary decision diagram (OBDD) is a graph representation of a Boolean function. It is observed that many practical Boolean functions are represented in feasible size. In this paper, the size of ordered binary decision diagrams representing threshold functions is studied. Two cases are treated: an ordering of variables is given in one case, and is adaptively chosen in the other case. We prove that upper bounds for both case are $\Omega(2^{n/2})$, where n is the number of input variables of threshold functions. We prove that a lower bound is $\Omega(2^{n/2})$ in the former case, and $\Omega(2^{\sqrt{n}/2})$ in the latter case.

1 Introduction

It is a very fundamental problem to represent and manipulate Boolean functions efficiently. Many data structures have been studied, such as truth tables, logic formulae, cube representations, logic circuits, etc. An ordered binary decision diagram (OBDD) [1] is a directed graph representing a Boolean function, and it is considered as one of the restricted types of branching programs. Properties of OBDD's are that many practical Boolean functions are represented in feasible size, Boolean operations are executed efficiently, and there exists a canonical representation when an ordering of variables is given. According to the good properties, OBDD's are widely used in many applications especially in computer-aided designs of logic circuits.

No data structure can represent all Boolean functions in polynomial size. It is natural to ask which functions can be or cannot be represented by polynomial size OBDD's.

The size of an OBDD representing a function may vary exponentially when the ordering of variables is changed.

For example, the size of an OBDD representing the output of an n -bit comparator with inputs $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $\mathbf{y} = (y_1, y_2, \dots, y_n)$ is linear when the ordering of input variables is $x_1 y_1 x_2 y_2 \cdots x_n y_n$, and is exponential when the ordering of input variables is $x_1 x_2 \cdots x_n y_1 y_2 \cdots y_n$. This shows that we should treat separately the case when an ordering of input variables is given and the case when any ordering of input variables can be chosen adaptively.

It is shown that the n -th bit of the output of an n -bit binary multiplier cannot be represented by an OBDD of polynomial size[2]. For some classes of Boolean functions, the size of OBDD's representing Boolean functions in the class is studied[3]. Tight lower bounds are proved for linear functions and symmetric functions. They are $\Theta(n)$ and $\Theta(n^2)$, respectively, where n is the number of input variables of the Boolean functions. For monotone functions and

self-dual functions, exponential lower bounds are proved by counting argument. Each lower bound is $\Omega(2^{n/2}/n)$ and $\Omega(2^n/n)$, respectively. An upper bound for all Boolean functions is $O(2^n/n)$ [4]. The bounds for self-dual functions and all Boolean functions are tight.

In this paper, we investigate the size of OBDD's representing threshold functions. A threshold function is a function whose output depends on whether a weighted sum of inputs is greater than a threshold value or not. Many theoretical approach have been made, such as the number of threshold functions[5], the maximum weight of threshold functions, realization of a Boolean function by a network of threshold functions[6].

It is shown that a lower bound and an upper bound for threshold function are $\Omega(n^2)$ and $O(2^n/n)$, respectively. It is noted in [7] that when weights are bounded by a polynomial of the number of variables, the size of an OBDD has a polynomial upper bound.

We discuss here two cases, which are the case when an ordering of variables is given, and the case when any ordering of variables can be chosen adaptively. In the case when an ordering of variables is given, we show that a lower bound is $\Omega(2^{n/2})$, and that the lower bound is tight to the constant factor. In the case when any ordering of variables can be chosen, we show that a lower bound is $\Omega(2^{\sqrt{n}/2})$.

This paper is organized as follows. In section 2, some definitions about threshold functions and OBDD's are given. In section 3, we investigate the size of OBDD's representing threshold functions, and prove some bounds on the size of an OBDD representing threshold functions. Conclusions and future works are noted in section 4.

2 Preliminaries

2.1 Threshold Function

Let $\mathbf{x} = (x_1, x_2, \dots, x_n) (\in \{0, 1\}^n)$ be input variables, $f(\mathbf{x})$ be a threshold function of n variables. $f(\mathbf{x})$ is defined by a set of n weights $\mathbf{w} = (w_1, w_2, \dots, w_n) (\in \mathbb{R}^n)$ and a threshold value $t (\in \mathbb{R})$ as follows.

$$f(\mathbf{x}) = \begin{cases} 0 & \text{if } \sum_{i=1}^n w_i \cdot x_i < t \\ 1 & \text{if } \sum_{i=1}^n w_i \cdot x_i \geq t \end{cases}$$

2.2 Ordered Binary Decision Diagram

An *OBDD* representing a Boolean function of n variables is a labeled, directed acyclic graph with only one source node. Each node has outdegree of 0 or 2. Sink nodes, whose outdegrees are 0, are called *constant nodes*, and are labeled by a constant value of 0 or 1. The other nodes, whose outdegrees are 2, are called *variable nodes*. Variable nodes are labeled by either of n input variables. Two outgoing edges of a variable node are labeled by 0 and 1, respectively. An edge labeled by 0 (1 resp.) is called a *0-edge* (*1-edge* resp.). The unique source node (indegree of 0) is called a *root node*.

Each path from the root node to a constant node has a length of n . We define the *level* of a node as the number of nodes in the path from the root node to the node. Note that all paths

from the root node to a node has the same length. Level i is called *higher*, (*lower* resp.) than level j if $i < j$ ($i > j$ resp.). All variable nodes in a level have the same label, and each variable is used as a label only once in a path from the root node to a constant node. Note that the ordering in which labels of nodes appear in a path is the same in all paths. This ordering is called a *variable ordering*. A variable ordering is defined by a permutation of input variables.

We assume each node has its unique node number. We call the node whose node number is i , in short, node i . We use a notation $\text{edge0}(i)$ ($\text{edge1}(i)$ resp.) to indicate the node number of 0-edge (1-edge resp.) successor of node i .

Each node of an OBDD represents a Boolean function. Let $\text{func}(i)$ be the Boolean function represented by node i . If node i is a constant node, $\text{func}(i)$ equals the label of the node. If node i is a variable node, the function is defined as follows by Shannon's expansion.

$$\begin{aligned} \text{func}(i) = & \quad \bar{x}_i \wedge \text{func}(\text{edge0}(i)) \\ & \vee x_i \wedge \text{func}(\text{edge1}(i)), \end{aligned}$$

where x_i is the label of the node i . The Boolean function represented by an OBDD is the Boolean function specified by the root node.

Two nodes a, b in the same level, which satisfy $\text{func}(a) = \text{func}(b)$ are called *equivalent nodes*. Two constant nodes are equivalent if their labels are same. Two variable nodes are equivalent if they have the same labels, equivalent 0-edge successors, and equivalent 1-edge successors. The *size* of an OBDD is the number of nodes in the OBDD. An OBDD is said to be *minimum*, if there is no equivalent node. Let W_i denote the number of nodes in level i . We call W_i the *width at level i* . Let $W = \max_{1 \leq i \leq n} W_i$. W is said to be the *width* of an OBDD.

An OBDD has following good properties. 1) There exists a canonical form for a Boolean function when a variable ordering is given. Furthermore, the canonical form is obtained from any OBDD's by merging equivalent nodes. 2) Many practical Boolean functions can be represented in feasible size if the variable ordering is properly chosen. 3) Many operations on OBDD's can be executed efficiently.

3 The Size of OBDD's Representing Threshold Functions

In this section, we argue the size of OBDD's representing threshold functions. We give an exponential upper bound, an exponential lower bound under a fixed variable ordering, and a super polynomial lower bound even if any variable ordering is allowed.

3.1 OBDD's Representing Threshold Functions

We introduce some properties of OBDD's representing threshold functions at first.

We define the *weight of a path* as the sum of weights for input variables corresponding to nodes whose outgoing 1-edges are in the path. There may be several paths from the root node to one node, and the weights of the paths may be different with one another. The set of the weights of paths from the root node to a node is called the *temporary sum set* of the node.

Let a and b be the smallest and greatest element in a temporary sum set of a node respectively. On a minimum OBDD, no number c which is contained in the temporary sum set of another node in the same level satisfies $a \leq c \leq b$. This means that the temporary sum set of a node can be represented by a closed interval from smallest element to greatest element in the set, and the intervals of nodes in a level never overlap one another.

If there is a path from the root node to a constant node whose weight is exactly the threshold value, each node i on the path is not equivalent to any node j which is in the same level and has less temporary sums. The reason is that by the assignment, $\text{func}(i) = 1$ and $\text{func}(j) = 0$.

3.2 Upper bound

We show some relations between widths of adjacent levels in the following lemma.

Lemma 1 On a minimum OBDD representing any n -variable threshold functions, the width of level i satisfies,

$$\begin{aligned} W_i &\leq 2W_{i-1}, \\ W_i &\leq 2W_{i+1} - 1. \end{aligned}$$

Proof The first relation is obvious one. This relation is true on the minimum OBDD's representing any Boolean function. On a minimum OBDD, temporary sum sets of nodes in a level are represented by closed intervals and none of the intervals overlaps. This means that nodes in a level is totally ordered.

Let a, b be nodes in the same level. Let $a \prec b$ denote that the temporary sums corresponding to node a are smaller than those of node b , and let $a \preceq b$ denote that the temporary sums corresponding to node a are the same with or smaller than those of node b . Then, if $a \prec b$, $\text{edge0}(a) \preceq \text{edge0}(b)$ and $\text{edge1}(a) \preceq \text{edge1}(b)$. Note that the case both $\text{edge0}(a) = \text{edge0}(b)$ and $\text{edge1}(a) = \text{edge1}(b)$ are satisfied means $a = b$, and never happens. Then we have $W_i \leq 2W_{i+1} - 1$. \square

An upper bound is presented here using Lemma 1.

Theorem 2 For a minimum OBDD representing a threshold function of n variables, the width W satisfies

$$W \leq \begin{cases} 2^{n/2} & (n: \text{ even}) \\ 2^{(n-1)/2} + 1 & (n: \text{ odd}). \end{cases}$$

Proof Let W_i be the width of an n -variable threshold function at the level i . Clearly, for $n \geq 1$, $W_1 = 1$ and $W_{n+1} = 2$. Lemma 1 shows $W_i \leq \min(2^{i-1}, 2^{n-i+1} + 1)$. \square

3.3 Lower Bounds

In this section, we show lower bounds for two cases. The first one is a tight lower bound in the case when the variable ordering is fixed.

Theorem 3 Let n be a positive even number. Let f be a threshold function of n variables whose weights w_i and threshold t are

$$\begin{aligned} w_i &= \begin{cases} 2^{i-1} & (1 \leq i \leq \lceil n/2 \rceil) \\ 2^{i-1-\lceil n/2 \rceil} & (\lceil n/2 \rceil + 1 \leq i \leq n) \end{cases} \\ t &= \sum_{i=1}^n w_i / 2. \end{aligned}$$

Let x_i be the variable corresponding the weight w_i . If the variable ordering is (x_1, x_2, \dots, x_n) , the width of the minimum OBDD representing f satisfies

$$W \geq \begin{cases} 2^{n/2} & (n: \text{ even}) \\ 2^{(n-1)/2} + 1 & (n: \text{ odd}). \end{cases}$$

Proof We prove the case when n is even at first. It is essential that all $(n/2)$ -bit binary number in which only one bit is 1 are used just twice as the weights, and that the threshold value is half of the sum of all weights.

We prove that the number of nodes in level $n/2 + 1$ is at least $2^{n/2}$. The weight of each path from the root node to level $n/2 + 1$ ranges all integers from 0 to $2^{n/2} - 1$, and differs from one another. Furthermore, for any nodes in level $n/2 + 1$, if assignments for levels lower than $n/2$ are chosen as $x_i = \overline{x_{i-n/2}}$ ($n/2 + 1 \leq i \leq n$), the weight of the corresponding path is exactly t , since $w_i = w_{i-n/2}$. This shows that at level $n/2 + 1$, any node is not equivalent to any other node which has smaller temporary sum of weights.

The case when n is odd is proved with a little more observation. The weights are the same with $(n-1)$ -variable function, except a $((n-1)/2 + 1)$ -bit binary number in which all but the most significant bit is 0 added.

We estimate the number of nodes in level $(n-1)/2 + 2$. The weight of each path from the root node to level $(n-1)/2 + 2$ ranges all integers from 0 to $2^{(n-1)/2+2} - 1$, and differs from one another. We prove that there are $2^{(n-1)/2+1}$ nodes for which there is a path from the root node to a constant node whose weight is exactly threshold value, and that there is at least one node from which there is no path to a constant node whose weight is at least threshold value. There are $2^{(n-1)/2+1}$ nodes whose assignments satisfy $x_{(n-1)/2} = \overline{x_{(n-1)/2+1}}$. For these nodes, the rest of assignment $x_i = \overline{x_{i-(n-1)/2-1}}$ ($((n+1)/2 + 2 \leq i \leq n-1)$ and $x_n = x_{(n-1)/2}$ makes the weight of the path exactly threshold value. On the other hand, from the node whose temporary sum is 0, there is no path to a constant node whose weight is at least threshold value. These $2^{(n-1)/2+1} + 1$ nodes cannot be the equivalent nodes. \square

Theorem 3 shows a lower bound on the size of OBDD's representing threshold functions under a fixed variable ordering. This lower bound matches the upper bound introduced in Theorem 2. However, this example can be expressed by a constant width OBDD when the variable ordering is $x_1 x_{1+n/2} x_2 x_{2+n/2} \dots x_{n/2} x_n$ (n : even), and $x_1 x_{1+(n-1)/2} x_2 x_{2+(n-1)/2} \dots x_{(n-1)/2} x_{n-1} x_n$ (n : odd).

We now investigate the case when any variable ordering can be chosen.

Theorem 4 Let k be a positive even number. Let g be the threshold function of k^2 variables whose weights $w_{i,j}$ and threshold t are

$$\begin{aligned} w_{i,j} &= 2^{i-1} + 2^{j+k-1} \quad (1 \leq i, j \leq k) \\ t &= \sum_{i=1}^k \sum_{j=1}^k w_{i,j} / 2. \end{aligned}$$

The width of the OBDD representing g satisfies $W \geq 2^{k/2}$ in any variable ordering.

Proof Weights of g are expressed in $2k$ -bit binary number, whose two bits are 1, and other bits are 0. One of the hot bit is in either of lower k bits, and the other hot bit is in either of higher k bits.

Let $x_{i,j}$ be a variable corresponding to the weight $w_{i,j}$, and $l(i,j)$ be the level of $x_{i,j}$. Let $I(l) = \{i \mid \exists j \text{ s.t. } l(i,j) \leq l\}$, $J(l) = \{j \mid \exists i \text{ s.t. } l(i,j) \leq l\}$. Let L be the minimum l satisfying either $|I(l)| = k/2$ or $|J(l)| = k/2$.

Firstly, we consider the case when $|I(L)| = k/2$. Let L' be minimum l satisfying $|J(l)| = k/2$. There are at least $2^{k/2}$ paths from the root node to level $L+1$ whose weights are different from one another, for $|I(L)| = k/2$ implies that there are, in levels higher than $L+1$, $k/2$ weights which differ in lower k bits. Let $v_1, v_2, \dots, v_{k/2}$ be the nodes which are the end points of the paths.

As in the proof of Theorem 3, we show that nodes $v_1, v_2, \dots, v_{k/2}$ cannot have the same temporary sum set, and that for each of $v_1, v_2, \dots, v_{k/2}$, there is a path from the root node to a constant node whose weight is exactly the threshold value.

The variables of levels from 1 to L are

$$x_{i,j} \quad (i \in I(L), j \in J(L'), l(i,j) \leq L).$$

For a given assignment for these variables, we describe an assignment for levels from $L+1$ to k^2 , which makes the weighted sum of inputs exactly the threshold value.

The assignment for

$$x_{i,j} \quad (i \in I(L), j \in J(L'), l(i,j) \geq L+1)$$

is 0.

Here, let $\pi(i), \rho(j)$ be permutation functions satisfying that $I(L) = \{\pi(i) \mid 1 \leq i \leq k/2\}$ and $J(L') = \{\rho(j) \mid 1 \leq j \leq k/2\}$. The assignment for

$$x_{i,j} \quad (i \notin I(L), j \in J(L'))$$

is chosen as $x_{\pi(i),j} = \overline{x_{\pi(i-k/2),j}}$. Note that weights for $x_{\pi(i),j}$ and $x_{\pi(i-k/2),j}$ are the same in higher k bits. The assignment for

$$x_{i,j} \quad (j \notin J(L'))$$

is chosen as $x_{i,\rho(j)} = \overline{x_{i,\rho(j-k/2)}}$. Note that weights for $x_{i,\rho(j)}$ and $x_{i,\rho(j-k/2)}$ are the same in lower k bits. After all, each bit of weights is added into the weight of the path exactly $k/2$ times. This is an assignment to make the weight of the path exactly the threshold value, for the threshold value is a half of sum of all weights.

The case when $|J(L)| = k/2$ can be treated in the same way. □

Theorem 4 shows that a lower bound on the size of an OBDD representing a n -variable threshold function is $\Omega(2^{\sqrt{n}/2})$ in any variable ordering.

4 Conclusion

In this paper, we studied the size of OBDD's representing threshold functions. We dealt with two cases. In the case when a variable ordering is given, we proved $2^{n/2}$ (n : even), $2^{(n-1)/2} + 1$ (n : odd) is the tight lower bound. In the case when any variable ordering is allowed, we proved a lower bound of $\Omega(2^{\sqrt{n}/2})$.

It is left as a future work to get a tight bound in the case when any ordering of variables is allowed.

Acknowledgment

We would like to express our appreciation to Associate Professor N. Takagi for his valuable comments. We would like to thank all the other members of Professor Yajima's Laboratory at Kyoto University for their useful discussions throughout this research.

References

- [1] R. E. Bryant: "Graph-Based Algorithms for Boolean Function Manipulation," IEEE Trans. Computers, C-35, 8, 677-691, (1986).
- [2] R. E. Bryant: "On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication," IEEE Trans. Computers, 40, 2, 205-213, (1991).
- [3] H. Sawada: "Computational Power of Binary Decision Diagrams," Master Thesis, KYOTO University, (1993).
- [4] H. T. Liaw and C. S. Lin: "On the OBDD-Representation of General Boolean," Functions, IEEE Trans. Computers, 41, 6, 661-664, (1992).
- [5] S. Yajima and T. Ibaraki: "A Lower Bound of the Number of Threshold Functions," IEEE Trans. Computers, EC-14, 6, 926-929, (1965).
- [6] S. Muroga: "Threshold Logic and its Applications," John Wiley & Sons, (1971).
- [7] N. Ishiura and S. Yajima: "A Class of Logic Functions Expressible by a Polynomial-Size Binary Decision Diagram," Proc. Synthesis and Simulation Meeting and International Interchange (SASIMI '90), 48-54, (1990).